

Desenvolvimento de Aplicativo WEB I

Prof. Me. Cleber Schroeder Fonseca

Curso de Tecnologia em Análise e Desenvolvimento de Sistemas
IFRS Rio Grande

2024



① Formulários

GET
POST

② REQUEST

③ Atividades

① Formulários

GET
POST

② REQUEST

③ Atividades

- Os formulários HTML são utilizados para o envio de informações para o servidor
- Os formulários encapsulam os dados como um vetor, cada posição do vetor é identificado com o nome do campo do formulário
- Exemplo:
 - Nome: `<input type="text" name="nome">`
 - Idade: `<input type="number" name="idade">`

```
form['nome'] = "Nome", form['idade'] = 10;
```

- O PHP recebe os dados do formulário de duas formas
 - Através dos métodos **GET** e **POST**
 - O formulário define o método na TAG form
 - `<form method="get" action="calcular.php"> ... </form>`
 - `<form method="post" action="calcular.php"> ... </form>`
- O PHP recebe o formulário dentro de um dos arrays
 - `$_GET` - caso o método do formulário seja get
 - `$_POST` - caso o método do formulário seja post

- O PHP recebe os dados do formulário de duas formas
 - Através dos métodos **GET** e **POST**
 - O formulário define o método na TAG form
 - `<form method="get" action="calcular.php"> ... </form>`
 - `<form method="post" action="calcular.php"> ... </form>`
- O PHP recebe o formulário dentro de um dos arrays
 - `$_GET` - caso o método do formulário seja get
 - `$_POST` - caso o método do formulário seja post



① Formulários

GET

POST

② REQUEST

③ Atividades

No PHP, o método GET é uma das maneiras de acessar os dados enviados pelos formulários HTML.

Os valores preenchidos nos campos do formulário são incluídos na URI.

Esses valores são automaticamente adicionados a variável superglobal “**`$_GET`**”.

Vamos ver um passo a passo como o método GET funciona:

- ➊ Vamos criar um formulário.
- ➋ Vamos identificar como os dados são enviados.
- ➌ Vamos ver como o PHP recebe e trata esses dados.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Formulários</title>
7 </head>
8 <body>
9     <form action="salvar.php" method="get">
10         Nome <input type="text" name="nome"><br>
11         <input type="submit" value="Salvar">
12     </form>
13 </body>
14 </html>
```

Quando o usuário clica no botão “salvar” os dados são encapsulados na URI, ao final do endereço afixado é colocado o caractere “?”.

Em seguida são colocados os dados da seguinte forma:
“nomeCampo=ValorPreenchido”. E sempre que forem enviados mais de um campo esses conjuntos chave=valor são separados pelo caractere “&”.

Exemplo:

`http://localhost/index.php?nome=Joao&idade=20`

```
1  <?php
2      echo $_GET['nome'];
3
4      if (isset($_GET['nome'])) {
5          $nome = $_GET['nome'];
6          echo "Olá, $nome!";
7      } else {
8          echo "Nome não informado.";
9      }
10  ?>
```

- **Simplicidade** - É muito simples de se entender, implementar e testar.
- **Cache** - O resultado pode ser armazenado no navegador ou por intermediários.
- **Navegação por URI** - Permite que usuários compartilhem resultados de operações simplesmente enviando a URI com os dados preenchidos.

- **Segurança** - Os dados são enviados pela URI, assim suscetíveis a ataques como *SQL Injection*, *Cross-site scripting* (XSS).
- **Restrição de tamanho** - As URIs tem restrição de tamanho dependo do navegador.
- **Exibição de dados** - Pelos dados serem enviados na URI qualquer pessoa pode ver o que foi enviado.
- **Navegação acidental** - Os usuários pode através do histórico acessar informações confidencial.



① Formulários

GET
POST

② REQUEST

③ Atividades

O método POST é amplamente utilizado quando trabalhamos com dados sensíveis (dados pessoais, senhas, etc).

Os valores preenchidos no formulário são “serializados” (agrupados na forma de um array) e são enviados ao servidor dentro do corpo dos pacotes HTTP.

Esses valores são então armazenados na variável superglobal “**`$_POST`**”.

Vamos ver novamente no passo a passo como funciona o método POST:

- ➊ Vamos criar um formulário.
- ➋ Vamos entender como os dados são enviados.
- ➌ Vamos ver como o PHP recebe e trata os dados.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Formulários</title>
7 </head>
8 <body>
9     <form action="salvar.php" method="post">
10         Nome <input type="text" name="nome"><br>
11         <input type="submit" value="Inserir">
12     </form>
13 </body>
14 </html>
```

Assim que o usuário clica em “salvar” os dados do formulário são serializados, mais ou menos como no método GET porém os dados são anexados ao cabeçalho da requisição HTTP.

```
1  <?php
2      echo $_POST['nome'];
3
4      if (isset($_POST['nome'])){
5          $nome = $_POST['nome'];
6          echo "Olá, $nome!";
7      } else {
8          echo "Nome não informado.";
9      }
10  ?>
```

- **Segurança dos dados** - Os dados não ficam visíveis. Portanto mais adequado a envio de informações sensíveis. Além disso se torna impossível monitoramento por cache.
- **Tamanho de dados ilimitado** - Não possui nenhuma limitação no tamanho de caracteres enviados.
- **Flexibilidade de dados** - Além do envio de texto também podem ser enviados JSON, XML, arquivos binários entre outros.
- **Compatibilidade com UTF-8** - Esse método lida muito bem com caracteres multi linguagens.

- **Complexidade** - Ele é mais complexo de testar, você precisa configurar o cabeçalho da forma correta para garantir que o servidor vai entender os dados.
- **Overhead de dados** - Como os dados são enviados no corpo da requisição ocorre de uma maior necessidade de banda de dados para a transmissão.
- **impossibilidade de compartilhamento de URIs** - Como os dados são enviados no corpo da requisição se torna impossível enviar para alguém o resultado de um envio do formulário.
- **Processamento do lado do servidor** - O processamento de dados pode ser mais complexo, especialmente quando é feito o upload de arquivos.

① Formulários

GET POST

② REQUEST

③ Atividades

- A variável `$_REQUEST` engloba os valores do `$_GET`, `$_POST` e do `$_COOKIE` (que será explicado posteriormente) em uma única variável.
- Isso quer dizer que ao usar o `$_REQUEST` todos os dados contidos nos arrays citados farão parte do array do request.
- Quando há conflitos entre dados de GET, POST e cookies com o mesmo nome, o PHP segue uma ordem de precedência para determinar qual valor deve ser usado. A ordem padrão é: POST tem precedência sobre GET, e GET tem precedência sobre cookies. Portanto, se houver uma variável com o mesmo nome em todas essas fontes, o valor do POST será utilizado.


```
1  <?php
2  echo $_REQUEST['nome'];
3
4  if (isset($_REQUEST['nome'])){
5      $nome = $_REQUEST['nome'];
6      echo "Olá, $nome!";
7  } else {
8      echo "Nome não informado.";
9  }
10 ?>
```



① Formulários

GET
POST

② REQUEST

③ Atividades

Atividade 1

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST


REQUEST

Atividades

Crie o formulário abaixo para:

- Aplicar uma operação nos dois valores
- Operações: soma, subtração, multiplicação, divisão e potenciação
- Crie uma página PHP para receber e calcular a operação selecionada

V1:

Soma 

V1:

Calcular

Atividade 2

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Escreva um algoritmo que peça para o usuário informar dois números. Então pergunte para ele a resposta da adição, subtração, multiplicação e divisão do primeiro pelo segundo número.

Então mostre na tela quantos acertos o usuário obteve.

Atividade 3

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Peça para o usuário informar sua idade. Então mostre na tela se ele é criança (menor de 13 anos), adolescente (entre 13 e 19 anos), adulto (entre 20 e 60 anos) ou idoso (mais de 60 anos).

Atividade 4

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Escreva um programa que peça para o usuário informar 10 números. Ao fim mostrar a média dos valores informados.

Atividade 5

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Escreva um programa que peça para o usuário informar quantos pacientes uma clínica pediátrica atendeu. Então abra um input para cada paciente para que sejam informadas as idades. Ao fim, mostre na tela a porcentagem de pacientes em cada uma das idades que possui pelo menos um paciente.

Atividade 6

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Escreva um programa que peça para o usuário inserir três números. No fim, escreva qual foi o maior e qual foi o menor.

Atividade 7

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Entrar com um número e informar se ele é divisível por 10, por 5, por 2 ou se não é divisível por nenhum destes

Atividade 8

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Entrar com 3 números e imprimi-los em ordem decrescente (suponha números diferentes)

Atividade 9

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

Crie um formulário em HTML com dois campos de texto para ler os valores do lado de um retângulo e um botão submit.

Crie uma página em PHP que receba os dados e exiba o valor da área e do perímetro do retângulo.

Crie um formulário em HTML com dois campos de formulário (um para o peso e outro para a altura de uma pessoa) e um botão submit. Crie uma página PHP que receba os dados e calcule o índice de massa corporal (IMC). O IMC é calculado da seguinte forma:

$$\text{imc} = \text{peso} / (\text{altura} * \text{altura})$$

Se o valor do IMC for maior do que 25 deverá ser exibida a mensagem “Você está acima do peso!”. Caso contrário exiba a mensagem “Você está saudável”.



INSTITUTO
FEDERAL
Rio Grande
do Sul

Desenvolvimento
de Aplicativo
WEB I

Prof. Me.
Cleber
Schroeder
Fonseca

Formulários

GET
POST

REQUEST

Atividades

MUITO OBRIGADO!

Cleber Schroeder Fonseca

<http://ifrs.edu.br/riogrande>

profcleberfonseca@gmail.com

cleber.fonseca@riogrande.ifrs.edu.br